

文章编号: 1001-0920(2011)05-0641-09

汽车同步装配线生产计划与调度集成优化

安玉伟^{1,2}, 严洪森¹

(1. 东南大学 a. 自动化学院, b. 复杂工程系统测量与控制教育部重点实验室, 南京 210096; 2. 黑龙江科技学院 数力系, 哈尔滨 150027)

摘要: 为克服传统的“自顶向下”方式下生产计划与调度不协调的缺陷, 针对汽车同步装配线, 构造了生产计划与调度集成优化混合整数规划模型, 并采用拉格朗日松弛法将其分解为批量计划及调度等子问题. 将调度子问题转化为与时间相关的旅行商问题, 并采用 dynasearch 算法求解. 对于拉格朗日对偶问题, 采用均衡方向策略法求解. 仿真实验结果验证了模型及算法的有效性.

关键词: 同步装配线; 生产计划与调度; 拉格朗日松弛; 均衡方向策略法; dynasearch 算法

中图分类号: TP391

文献标识码: A

Integrated optimization for production planning and scheduling on synchronous automobile assembly line

AN Yu-wei^{1,2}, YAN Hong-sen¹

(1a. School of Automation, 1b. Key Laboratory of Measurement and Control of CSE of Ministry of Education, Southeast University, Nanjing 210096, China; 2. Department of Mathematics and Mechanics, Heilongjiang Institute of Science and Technology, Harbin 150027, China. Correspondent: AN Yu-wei, E-mail: anyuwei7@163.com)

Abstract: To overcome the incoordination between production plan and its schedule in the classic from-top-to-down approach, a mixed integer programming model of integrated production planning and scheduling is presented for synchronous automobile assembly line. A Lagrangian relaxation method is developed for the proposed model which is decomposed into planning and scheduling sub-problems, etc. The scheduling sub-problem is modeled as a time-dependent traveling salesman problem (TDTSP) which is solved by using a dynasearch algorithm. The average direction strategy is employed to solve the Lagrangian dual problem, and simulation results show the effectiveness of the proposed model and algorithm.

Key words: synchronous assembly line; production planning and scheduling; Lagrangian relaxation; average direction strategy method; dynasearch algorithm

1 引言

生产计划与调度分属于制造系统中不同的决策层次, 同时它们之间又存在紧密相关的联系, 生产计划层的决策是调度层的优化目标. 然而, 在现行的生产管理决策中计划和调度决策是分开的, 常采用“自顶向下”的策略. 这种决策方法的主要问题是由于计划层忽略了调度层的详细约束, 造成依据计划目标所制定的调度往往不可行^[1].

理想的决策方法是将计划与调度集成起来协同优化^[2-3]. 主要方法有递阶分解法、迭代法及全空间法(也称整体法)^[4-5]. 递阶分解法将问题分解为上、下两个层次子问题, 上层子问题确定生产目标, 即计划

子问题; 下层子问题为详细的调度问题. 上层子问题的决策被作为参数传递给下层调度子问题. 递阶分解法的主要缺点是得到的计划不可行, 或得到了一个性能指标较差的调度^[4]. 迭代法与递阶分解法的区别在于信息流是双向的, 即不仅计划层的决策传递给调度层, 同时调度层的信息也反馈给计划层. 计划与调度交替求解, 最后使计划与调度同时得到优化. 这种方法的局限性在于一般只能收敛于局部解. 全空间法将详细的调度约束作为计划问题的资源约束, 同时将调度的性能指标转化为生产费用结合到计划问题的目标函数中, 建立起计划与调度问题的统一模型, 模型包含了生产过程的所需信息. 但由于所建立的模型比

收稿日期: 2010-01-28; 修回日期: 2010-07-12.

基金项目: 国家863计划项目(2007AA04Z112); 国家自然科学基金项目(60934008, 50875046).

作者简介: 安玉伟(1968—), 男, 副教授, 博士生, 从事生产计划与调度、运筹学等研究; 严洪森(1957—), 男, 教授, 博士生导师, 从事生产计划与调度、知识化制造等研究.

较复杂,需要研究者提出较好的求解方法,其中各种智能算法(如禁忌搜索^[6],遗传算法^[7-8]等)是常用的方法。

文献[6]给出了汽车装配车间生产计划与调度集成优化模型,并提出采用3种新的禁忌搜索法进行求解,通过调度仿真确定调度的性能指标.[9]研究了流水车间批量计划与调度的集成优化问题,采用协同进化算法与遗传算法并行混合搜索策略进行求解,得到一个近优的生产计划与调度解,但如何建立有效的协调机制是比较困难的.[10]针对多品种批量生产的流水车间环境,建立了计划与调度同时优化的两层混合整数规划模型,通过综合运用遗传算法与调度规则,确定了生产批量与投产顺序。

全空间法所建立的模型多为块状结构(即模型的约束集可分为较简单的约束集与复杂的约束集),求解的另外一种途径是应用拉格朗日松弛(LR)法^[4,11].本文研究了汽车同步装配线的生产计划与调度集成优化问题.首先讨论了同步装配线的特点,然后建立了耦合详细调度约束的计划与调度同时优化的混合整数规划模型,本质上属于全空间法.采用LR法将模型分解为一系列的计划子问题与调度子问题,调度子问题可转化为一种与时间相关的旅行商问题(TDTSP),并应用dynasearch算法求解TDTSP.

2 问题描述

2.1 同步流水装配线

本文的研究背景与文献[6]相同,均以南京某汽车总装厂的装配线为研究对象.设一条装配线共有 m 个装配工位,计划区间 H 内依据装配订单共需要装配 i 种汽车(也称为工件),且第 i 种汽车需 d_i 辆.每辆汽车依次以顺序 $i = 1, 2, \dots, m$ 在 m 个工位(机器)上装配,同一时刻同一工位最多只能装配一辆汽车,每辆汽车任何时刻至多能在一个工位上装配(也称为加工),汽车在每个工位上的排序都是一致的,即排列排序.装配线上各工位的装配操作完成时间可以是不同的,所有工位装配操作完成后,装配线上的汽车同步移动到下一个工位.当工位之间的空间有限或工件体积较大时,装配线一般是无缓冲区的,采用同步移动方式可以有效地避免由于同时加工与运输造成的工件堵塞现象^[12-14].这种装配线称为同步流水装配线.如果在作业车间的生产线上采用这种同步移动方式,则成为文献[13]所研究的同步流水作业车间(SFS).

相邻两次移动时间间隔称为一个装配节拍.设 C^k 为第 k 个装配节拍, n 个工件在由 m 个装配工位组成的同步流水装配线上装配,需要 $n+m-1$ 个装配节

拍.文献[13]证明了当 $m \geq 3$ 时,极小化makespan的SFS的调度问题是NP-hard的.

设 p_{ij} 为第 i 种汽车在第 j 个装配工位上装配所需要的时间,图1中^[13-14],3个工件依次在M1, M2, M3三个工位上加工,则

$$C^1 = p_{11}, C^2 = \max\{p_{12}, p_{21}\},$$

$$C^3 = \max\{p_{13}, p_{22}, p_{31}\},$$

$$C^4 = \max\{p_{23}, p_{32}\}, C^5 = p_{33}.$$

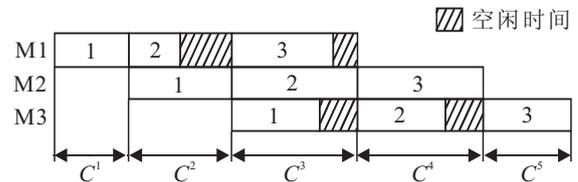


图1 同步流水装配线

2.2 最小批量约束及同步流水装配线排序问题

在某些生产环境中,由于技术上的限制或由于生产成本的制约及管理上的考虑,产品一旦确定生产,其生产量必须大于一个给定的批量^[15-18].这种限制称之为最小批量约束.本文根据所研究的生产线的特点,假设第 i 种汽车的最小批量 $mls_i \geq m$.

设 $\pi = (\pi(0), \pi(1), \dots, \pi(n), \pi(n+1))$ 表示 n 种汽车的一个装配排序, $\pi(k) = i(i, k = 1, 2, \dots, n)$ 表示第 i 种汽车排在第 k 位置上装配.为叙述方便,称 $\pi(k)$ 为第 k 子批,其中 $\pi(0) = \pi(n+1) = 0$ 为虚拟汽车,并且规定 $p_{\pi(0),j} = p_{\pi(n+1),j} = 0$, $s_{\pi(0),j} = s_{\pi(n+1),j} = 0$, $j = 1, 2, \dots, m$,其中 $s_{i,j}$ 为第 i 种汽车在第 j 个装配工位上的准备时间.设 Π 为包含虚拟汽车在内的所有可能排序集合. $x_i v$ 为第 i 种汽车的生产批量, $C^{\pi(k),l}$ 为子批 $\pi(k)$ 的第 l 个装配节拍所用的时间,则

$$C^{\pi(k),l} = \begin{cases} \max_{\substack{1 \leq r \leq l \\ l+1 \leq j \leq m}} \{\tilde{p}_{\pi(k),l,r}, p_{\pi(k-1),j}\}, & l = 1, \dots, m-1; \\ \max_{1 \leq r \leq m} \{\tilde{p}_{\pi(k),l,r}\}, & l = m, \dots, x_{\pi(k)}; \\ \max_{\substack{1 \leq r \leq l - x_{\pi(k)} \\ l - x_{\pi(k)} + 1 \leq j \leq m}} \{\tilde{p}_{\pi(k+1),l-x_{\pi(k)},r}, p_{\pi(k),j}\}, & \\ & l = x_{\pi(k)} + 1, \dots, x_{\pi(k)} + m - 1. \end{cases} \quad (1)$$

其中

$$\tilde{p}_{\pi(k),l,r} = \begin{cases} p_{\pi(k),r} + s_{\pi(k),r}, & r = l; \\ p_{\pi(k),r}, & r \neq l. \end{cases}$$

在式(1)中,当 $m \leq l \leq x_{\pi(k)}$ 时, $C^{\pi(k),l}$ 的值仅与子批 $\pi(k)$ 在各机器上的装配时间有关,而与其他子批无关,记 $B_{\pi(k)} = \sum_{l=m}^{x_{\pi(k)}} C^{\pi(k),l}$,则有

$$B_{\pi(k)} = (x_{\pi(k)} - m)P_{\pi(k)} + \max_{1 \leq r \leq m} \{\tilde{p}_{\pi(k),m,r}\},$$

其中

$$P_{\pi(k)} = \max_{1 \leq j \leq m} \{p_{\pi(k),j}\}. \quad (2)$$

当 $1 \leq l \leq m-1$ 时, $C^{\pi(k),l}$ 的值与子批 $\pi(k)$, $\pi(k-1)$ 有关, 它们的和 $\sum_{l=1}^{m-1} C^{\pi(k),l}$ 表示相邻两子批 $\pi(k)$, $\pi(k-1)$ 的耦合费用, 记为

$$\text{Cop}_{\pi(k-1),\pi(k)} = \sum_{l=1}^{m-1} \max_{1 \leq r \leq l} \{\tilde{p}_{\pi(k),l,r}, p_{\pi(k-1),j}\}, \quad (3)$$

记子批 $\pi(k)$ 在生产线上总加工时间为 $t_{\pi(k)}$, 有

$$t_{\pi(k)} = C^{\pi(k),1} + \dots + C^{\pi(k),x_{\pi(k)}+m-1} =$$

$$\text{Cop}_{\pi(k-1),\pi(k)} + B_{\pi(k)} + \text{Cop}_{\pi(k),\pi(k+1)}. \quad (4)$$

记汽车 i 开始加工及完工时间分别为 s_i 和 C_i , 可以得到各子批的开工与完工时间为

$$s_{\pi(0)} = C_{\pi(0)} = 0; \quad (5)$$

$$s_{\pi(1)} = 0, C_{\pi(1)} = t_{\pi(1)}; \quad (6)$$

$$s_{\pi(k)} = C_{\pi(k-1)} - \text{Cop}_{\pi(k-1),\pi(k)}, k = 2, 3, \dots, n; \quad (7)$$

$$C_{\pi(k)} = s_{\pi(k)} + t_{\pi(k)}, k = 1, 2, \dots, n; \quad (8)$$

$$C_{\pi(n+1)} = C_{\pi(n)}. \quad (9)$$

式(7)表示相邻两个子批之间的开工与完工时间存在耦合关系. 根据式(4)和(8)依次递推可以得到

$$C_{\pi(k)} = \sum_{l=1}^{k+1} \text{Cop}_{\pi(l-1),\pi(l)} + \sum_{l=1}^k B_{\pi(l)}, \quad k = 1, 2, \dots, n. \quad (10)$$

2.3 生产计划与调度集成优化模型

设 h_i^+ 为超产 1 辆第 i 种汽车的库存费用, h_i^- 为欠产 1 辆第 i 种汽车所受到的惩罚费用, 通常假设欠产惩罚费用函数为线性函数^[2,6-8]. 在实际生产中, 当欠产超过一定额度时, 单位欠产惩罚费用将增大^[19]. 本文欠产惩罚费用函数为分段线性函数, 即

$$h_i^-(x) = \begin{cases} a_i x, & x < r_i; \\ b_i x - (b_i - a_i)r_i, & x \geq r_i. \end{cases}$$

式中: $a_i < b_i$ 为第 i 种汽车的不同欠产程度单位惩罚成本, r_i 为不同惩罚成本的分段点. 建立如下生产计划与调度集成优化的混合整数非线性规划模型 MINLP:

$$\begin{aligned} \min Z(Q) = & \\ & \sum_{i=1}^n \left(h_i^-(I_i^-) + h_i^+ I_i^+ + c_i x_i + \right. \\ & \left. \sum_{j=1}^m s c_{ij} y_i + e_i E_i + w_i T_i \right) + \sum_{j=1}^m g c_j s l_j. \quad (11) \end{aligned}$$

$$\text{s.t. } x_i + I_i^- - I_i^+ = d_i, i = 1, 2, \dots, n; \quad (12)$$

$$\sum_{i=1}^n p_{ij} x_i + \sum_{i=1}^n f_{ij} y_i + s l_j = \tau_j, \quad j = 1, 2, \dots, m; \quad (13)$$

$$0 \leq x_i \leq M y_i, i = 1, 2, \dots, n; \quad (14)$$

$$x_i \geq m l s_i y_i, i = 1, 2, \dots, n; \quad (15)$$

$$T_{\pi(k)} \geq C_{\pi(k)} - R_{\pi(k)}, k = 1, 2, \dots, n; \quad (16)$$

$$E_{\pi(k)} \geq R_{\pi(k)} - C_{\pi(k)}, k = 1, 2, \dots, n; \quad (17)$$

$$C_{\pi(k)} = \sum_{l=1}^{k+1} \text{Cop}_{\pi(l-1),\pi(l)} + \sum_{l=1}^k B_{\pi(l)}, \quad k = 1, 2, \dots, n; \quad (18)$$

$$\pi \in \Pi; \quad (19)$$

$$I_i^-, I_i^+, x_i, T_i, E_i \geq 0, i = 1, 2, \dots, n; \quad (20)$$

$$y_i \in \{0, 1\}, i = 1, 2, \dots, n; \quad (21)$$

$$s l_j \geq 0, j = 1, 2, \dots, m. \quad (22)$$

模型中已在前文中出现的符号不再解释, 这里只对新出现的符号加以说明: c_i 为第 i 种汽车的单位生产成本, $s c_{ij}$ 为第 i 种汽车在装配工位 j 上的准备成本, $g c_j$ 为与第 j 个装配工位闲置有关的成本, e_i 和 w_i 分别为拖期交货及提前交货单位惩罚成本; f_{ij} 为第 i 种汽车在装配工位 j 上的准备时间, $s l_j$ 为装配工位 j 上的空闲时间, τ_j 为计划区间内装配工位 j 上的可用时间; M 为一个大的正数; R_i 第 i 种汽车的交货期. 决策变量 x_i 为计划区间内第 i 种汽车的生产批量, I_i^- 为第 i 种汽车的欠产量, I_i^+ 为第 i 种汽车的超产量. y_i 表示第 i 种汽车是否生产, 其值当 $x_i > 0$ 时为 1, 否则为 0; T_i 为第 i 种汽车的交货拖期; E_i 为第 i 种汽车的交货提前期; π 为包含虚拟汽车在内的一个排序. 记 $Q = (I^-, I^+, x, y, sl, \pi, E, T) = (I^-, I^+, q)$ 为决策变量矩阵.

式(11)为优化目标函数, 包括欠产惩罚费用、库存费用、生产费用、生产准备费用、装配工位闲置成本、提前交货及拖期交货需要支付的费用. 式(12)表示物料平衡, 式(13)为能力约束, 式(14)保证了只有当批量大于零时才进行生产准备, 式(15)为最小批量限制, 式(16)及(17)分别为交货拖期及交货提前期满足的条件, 式(18)表示子批 $\pi(k)$ 的完工时间, 式(19)~(22)表示变量约束条件.

3 拉格朗日松弛法

LR 法目前仍是求解大规模优化问题常用的有效方法^[20]. 其基本原理是避开直接处理优化模型中的复杂约束, 将这些复杂约束作为惩罚项加到目标函数

中,转化为一个目标函数值为原问题的下界(对于目标函数求极小问题)的新问题. 新问题可以分解为一系列易于求解的子问题,并通过求解拉格朗日对偶问题而逐步逼近原问题的最优解,进而获取近优解.

3.1 MINLP 模型的松弛与分解

从 2.3 节的 MINLP 模型可以看出,把约束 (18) 代入 (16) 和 (17) 后,只有约束 (16) 和 (17) 耦合着批量与排序变量,引入非负拉格朗日乘子 $\lambda = (\lambda_1, \dots, \lambda_n)$, $\mu = (\mu_1, \dots, \mu_n)$,将约束 (16) 和 (17) 松弛到目标函数中,得到拉格朗日函数

$$L(\lambda, \mu) = \min_Q \tilde{L}(Q, \lambda, \mu).$$

其中

$$\begin{aligned} \tilde{L}(Q, \lambda, \mu) = & Z(Q) + \sum_{k=1}^n (\lambda_{\pi(k)}(C_{\pi(k)} - R_{\pi(k)} - \\ & T_{\pi(k)}) + \mu_{\pi(k)}(R_{\pi(k)} - C_{\pi(k)} - E_{\pi(k)})). \end{aligned} \quad (23)$$

记

$$\tilde{c}_{\pi(k)} = c_{\pi(k)} + \sum_{s=k}^n (\lambda_{\pi(s)} - \mu_{\pi(s)})P_{\pi(k)}, \quad (24)$$

对式 (23) 进行分类合并,得到如下形式:

$$\begin{aligned} \tilde{L}(Q, \lambda, \mu) = & \sum_{i=1}^n (h_i^-(I_i^-) + h_i^+I_i^+ + \sum_{j=1}^m sc_{ij}y_i) + \\ & \sum_{k=1}^n \tilde{c}_{\pi(k)}x_{\pi(k)} + \sum_{j=1}^m gc_jsl_j + \\ & \sum_{k=1}^n \sum_{l=1}^{k+1} (\lambda_{\pi(k)} - \mu_{\pi(k)})\text{Cop}_{\pi(l-1),\pi(l)} + \\ & \sum_{k=1}^n \sum_{s=k}^n (\lambda_{\pi(s)} - \mu_{\pi(s)}) \left(\max_{1 \leq r \leq m} \{\tilde{p}_{\pi(k),m,r}\} - mP_{\pi(k)} \right) - \\ & mP_{\pi(k)} + \sum_{k=1}^n (w_{\pi(k)} - \lambda_{\pi(k)})T_{\pi(k)} + \\ & \sum_{k=1}^n (e_{\pi(k)} - \mu_{\pi(k)})E_{\pi(k)} + \\ & \sum_{k=1}^n (\mu_{\pi(k)} - \lambda_{\pi(k)})R_{\pi(k)}. \end{aligned} \quad (25)$$

拉格朗日松弛问题可以分解为批量计划子问题 P_{PP} , 提前期子问题 P_E , 拖期子问题 P_T 和子批排序(调度)子问题 P_{SQ} . 其中批量计划子问题 P_{PP}

$$\begin{aligned} Z_{PP}(\lambda, \mu) = & \min_{I, x, y} F, \\ F = & \sum_{i=1}^n (h_i^-(I_i^-) + h_i^+I_i^+ + \sum_{j=1}^m sc_{ij}y_i) + \end{aligned}$$

$$\sum_{k=1}^n \tilde{c}_{\pi(k)}x_{\pi(k)} + \sum_{j=1}^m gc_jsl_j; \quad (26)$$

s.t. 式 (12)~(15), (20)~(22).

提前期子问题 P_E

$$Z_E(\mu) = \min_{E_{\pi(k)} \geq 0} \sum_{k=1}^n (e_{\pi(k)} - \mu_{\pi(k)})E_{\pi(k)}. \quad (27)$$

拖期子问题 P_T

$$Z_T(\mu) = \min_{T_{\pi(k)} \geq 0} \sum_{k=1}^n (w_{\pi(k)} - \lambda_{\pi(k)})T_{\pi(k)}. \quad (28)$$

调度子问题 P_{SQ}

$$\begin{aligned} Z_{SQ}(\lambda, \mu) = & \min_{\pi} \tilde{Z}_{SQ}(\pi, \lambda, \mu) = \\ & \min_{\pi} \sum_{k=1}^n \left[\sum_{l=1}^{k+1} (\lambda_{\pi(k)} - \mu_{\pi(k)})\text{Cop}_{\pi(l-1),\pi(l)} + \right. \\ & \left. \sum_{s=k}^n (\lambda_{\pi(s)} - \mu_{\pi(s)}) \left(\max_{1 \leq r \leq m} \{\tilde{p}_{\pi(k),m,r}\} - mP_{\pi(k)} \right) \right]. \end{aligned} \quad (29)$$

则

$$\begin{aligned} L(\lambda, \mu) = & Z_{PP}(\lambda, \mu) + Z_E(\mu) + Z_T(\lambda) + \\ & Z_{SQ}(\lambda, \mu) + \sum_{k=1}^n (\mu_{\pi(k)} - \lambda_{\pi(k)})R_{\pi(k)}. \end{aligned} \quad (30)$$

上述拉格朗日松弛的对偶问题为

$$\text{LD} = \max\{L(\lambda, \mu) : \lambda \geq 0, \mu \geq 0\}.$$

从式 (24) 和 (26) 可以看出,批量计划子问题 P_{PP} 的目标函数与子批的排序结果有关,求解调度子问题 P_{SQ} 后得到的排序结果输入给子问题 P_{PP} 的目标函数. 因此,子问题 P_{PP} 和 P_{SQ} 的求解顺序是首先求解子问题 P_{PP} , 然后再求解子问题 P_{SQ} . 这一点与通常的拉格朗日松弛法中各子问题是相互独立求解的情形不同. 另外,这里虽然是顺序求解子问题 P_{PP} 和 P_{SQ} ,但这两个子问题的目标函数是通过拉格朗日乘子相关联的,与前面提到的递阶法不同. 需要指出的是,提前期子问题及拖期子问题在形式上与调度有关,但其求解结果不依赖于调度,具体讨论请见第 3.2.3 节. 图 2 给出了拉格朗日松弛分解框架.

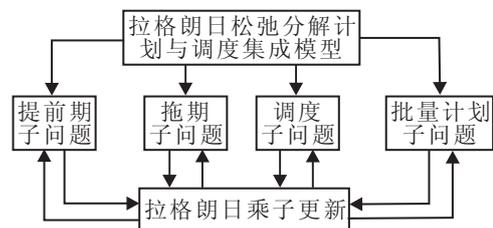


图 2 拉格朗日松弛分解框架

3.2 子问题求解方法

3.2.1 批量计划子问题 P_{PP} 求解

对于给定的子批排序 π , 显然有 $\sum_{k=1}^n \tilde{c}_{\pi(k)} x_{\pi(k)} = \sum_{k=1}^n \tilde{c}_i x_i$, 批量计划子问题 P_{PP} 的目标函数是非凸分段线性函数, 可将其转化为线性函数, 从而子问题 P_{PP} 转化为混整数线性规划.

定理1 子问题 P_{PP} 等价于如下混合整数线性规划问题 \tilde{P}_{PP} :

$$\begin{aligned} \min_{I, x, y} \tilde{F} = & \sum_{i=1}^n (b_i I_i^{1-} + (a_i - b_i) I_i^{2-} + \\ & h_i^+ I_i^+ + \sum_{j=1}^m s c_{ij} y_j + \tilde{c}_i x_i) + \sum_{j=1}^m g c_j s l_j. \quad (31) \\ \text{s.t. } & I_i^{1-} + x_i - I_i^+ = d_i, \quad i = 1, 2, \dots, n; \quad (32) \\ & I_i^{1-} \geq I_i^{2-}, \quad i = 1, 2, \dots, n; \quad (33) \\ & I_i^{2-} \leq r_i, \quad i = 1, 2, \dots, n; \quad (34) \\ & I_i^{1-}, I_i^{2-}, x_i, T_i, E_i \geq 0, \quad i = 1, 2, \dots, n. \quad (35) \end{aligned}$$

并满足约束(13)~(15)及(21)~(22).

证明 记

$$\theta(I^+, q_1) = \sum_{i=1}^n (h_i^+ I_i^+ + \tilde{c}_i x_i + s_i y_i) + \sum_{j=1}^m g c_j s l_j.$$

首先证明子问题 P_{PP} 的最优解也为 \tilde{P}_{PP} 的最优解. 设子问题 P_{PP} 的最优解为 (I^{+*}, I^{-*}, q_1^*) , 则其最优目标值为 $F = \sum_{i=1}^n h_i^-(I_i^{-*}) + \theta(I^{+*}, q_1^*)$, 下面分两种情况讨论:

1) 若 $I_i^{-*} \geq r_i$, 则 $h_i^-(I_i^{-*}) = b_i I_i^{-*} + (a_i - b_i) r_i$, 对于子问题 \tilde{P}_{PP} , 取 $I_i^{1-*} = I_i^{-*}$, $I_i^{2-*} = r_i$, 易证 $(I^{1-*}, I^{2-*}, q_1^*)$ 为 \tilde{P}_{PP} 的最优解, 且目标函数值为 $\tilde{F} = \sum_{i=1}^n (b_i I_i^{-*} + (a_i - b_i) r_i) + \theta(I^{+*}, q_1^*)$.

2) 若 $I_i^{-*} < r_i$, 则 $h_i^-(I_i^{-*}) = a_i I_i^{-*}$, 取 $I_i^{1-*} = I_i^{-*}$, 易证 $(I^{1-*}, I^{2-*}, q_1^*)$ 为 \tilde{P}_{PP} 的最优解, 且 $\tilde{F} = \sum_{i=1}^n a_i I_i^{-*} + \theta(I^{+*}, q_1^*)$.

然后证明子问题 \tilde{P}_{PP} 的最优解也为 P_{PP} 的最优解. 设 \tilde{P}_{PP} 的最优解为 $(I^{1-*}, I^{2-*}, q_1^*)$, 同样分两种情况讨论:

1) 若 $I_i^{1-*} \geq r_i$, 因为在目标函数中 I_i^{2-*} 的系数小于零, 则根据约束(33),(34)及优化问题是求极小值, 问题 \tilde{P}_{PP} 的最优解中必有 $I_i^{2-*} = r_i$, 目标函数值为 $\tilde{F} = \sum_{i=1}^n (b_i I_i^{1-*} + (a_i - b_i) r_i) + \theta(I^{+*}, q_1^*)$. 令 I_i^{-*}

$= I_i^{1-*}$, 易证 (I^{+*}, I^{1-*}, q_1^*) 为 P_{PP} 的最优解, 且 $F = \sum_{i=1}^n (b_i I_i^{1-*} + (a_i - b_i) r_i) + \theta(I^{+*}, q_1^*)$.

2) 若 $I_i^{1-*} < r_i$, 则由 \tilde{P}_{PP} 的结构特点, 在最优解中, 有 $I_i^{2-*} = I_i^{1-*}$, 且 $\tilde{F} = \sum_{i=1}^n a_i I_i^{1-*} + \theta(I^{+*}, q_1^*)$. 取 $I^{-*} = I^{1-*}$ 易证 (I^{+*}, I^{1-*}, q_1^*) 为 P_{PP} 的最优解, 且 $F = \sum_{i=1}^n a_i I_i^{1-*} + \theta(I^{+*}, q_1^*)$. \square

至此将非凸混合整数规划 P_{PP} 问题转化为混合整数线性规划 \tilde{P}_{PP} 问题, \tilde{P}_{PP} 问题可用分枝定界法或割平面法等精确优化方法求解.

3.2.2 提前期子问题 P_E 及拖期子问题 P_T 的求解

式(27)和(28)中的目标函数均是对全体子批求和, 且是无约束优化问题, 由 $\pi(k)$ 的定义, 可以得出

$$\begin{aligned} \sum_{k=1}^n (e_{\pi(k)} - \mu_{\pi(k)}) E_{\pi(k)} &= \sum_{i=1}^n (e_i - \mu_i) E_i, \\ \sum_{k=1}^n (w_{\pi(k)} - \lambda_{\pi(k)}) T_{\pi(k)} &= \sum_{i=1}^n (w_i - \lambda_i) T_i. \end{aligned}$$

这样可以确定 P_E 的最优解为: 当 $e_{\pi(k)} - \mu_{\pi(k)} \geq 0$ 时, $E_{\pi(k)} = 0$; 当 $e_{\pi(k)} - \mu_{\pi(k)} < 0$ 时, $E_{\pi(k)} = \infty$. 同理, 拖期调度子问题 P_T 的最优解为: 当 $w_{\pi(k)} - \lambda_{\pi(k)} \geq 0$ 时, $T_{\pi(k)} = 0$; 当 $w_{\pi(k)} - \lambda_{\pi(k)} < 0$ 时, $T_{\pi(k)} = \infty$.

3.2.3 调度子问题 P_{SQ} 的求解策略

调度子问题 P_{SQ} 的目标函数经过适当的改写, 可将其转化为一种特殊形式的旅行商(TSP)问题, 具体过程如下:

设 $\varphi_{\pi(k)} = \lambda_{\pi(k)} - \mu_{\pi(k)}$, $k = 1, 2, \dots, n$, 并规定 $\varphi_{\pi(0)} = 0$, 则记 $u_{\pi}^k = \sum_{s=k}^n \varphi_{\pi(s)}$, $k = 0, 1, \dots, n$. 注意到当 $k = 0$ 时, 由于规定虚拟子批在各个装配工位的加工时间均为零, 所以

$$\begin{aligned} & [\text{Cop}_{\pi(k), \pi(k+1)} + \max_{1 \leq r \leq m} \{\tilde{p}_{\pi(k), m, r}\} - \\ & m P_{\pi(k)}] |_{k=0} = \text{Cop}_{\pi(0), \pi(1)}. \quad (36) \end{aligned}$$

令

$$\begin{aligned} d_{\pi(k), \pi(k+1)} = & \text{Cop}_{\pi(k), \pi(k+1)} + \max_{1 \leq r \leq m} \{\tilde{p}_{\pi(k), m, r}\} - m P_{\pi(k)}, \\ & k = 0, 1, \dots, n, \quad (37) \end{aligned}$$

且 $u_{\pi}^0 = u_{\pi}^1$, 则调度子问题 P_{SQ} 重新表述为 \tilde{P}_{SQ} , 即

$$\begin{aligned} Z_{SQ}(\lambda, \mu) = & \min_{\pi} \sum_{k=0}^n u_{\pi}^k d_{\pi(k), \pi(k+1)} \\ \text{s.t. } & \pi \in \Pi. \quad (38) \end{aligned}$$

为了求解调度子问题 \tilde{P}_{SQ} , 构造网络 $G = (V, E)$, 顶点集 $V = \{0, 1, 2, \dots, n, n+1\}$ 为包含虚拟工件(汽

车)在内的工件集, $E = \{(i, j) | i, j \in V, i \neq j\}$ 为一弧集. 前边定义的汽车(工件)加工序列 $\pi = (\pi(0), \pi(1), \pi(2), \dots, \pi(n), \pi(n+1))$, 在这里可以理解为从虚拟顶点出发遍历所有顶点且每一个顶点只能经过一次, 又回到虚拟顶点的一条哈密顿圈. 定义任意两个节点 i 和 j 之间的距离为 d_{ij} , 如果节点 i 和 j 对应的弧排在哈密顿圈第 $k, k+1$ 个位置, 即 $\pi(k) = i, \pi(k+1) = j$, 则赋予权重(旅行费用) u_{π}^k . 从而调度子问题 \tilde{P}_{SQ} 可以理解为一种特殊的旅行商问题.

由于 TDTSP 为 NP-hard 的, 分支定界等优化方法在求解较大规模问题时, 需要较多时间消耗, 只能用于求解中小规模问题的最优解. 根据 TDTSP 本质上是一种置换问题, 及 dynasearch 算法在求解该类问题上的优势^[21], 本文采用 dynasearch 算法求解 TDTSP. Congram^[21-22]提出的 dynasearch 算法在研究问题结构特征的基础上, 通过利用常用邻域(如 swap, insert 等)的一系列相互独立的组合进行搜索, 能够在多项式时间内完成指数数量级的邻域搜索. dynasearch 算法的本质是一种动态规划方法, 已在求解单机调度、车辆调度等问题上取得了较好的效果^[23]. 文献[24]针对酸轧生产调度问题, 提出一种嵌入强化 dynasearch 算法的禁忌搜索(TS)混合算法. 该算法可以有效地跳出局部最优限制, 并通过降低搜索空间, 使算法加速.

给定初始工件排序 $\pi \in \Pi$, π_k 表示部分工件 $\pi(1), \dots, \pi(k)$ 经过一系列独立的互换操作(swap)得到的部分最优排序, 即 $\pi(1), \dots, \pi(k)$ 之间的最短路径. 注意 π_k 中的最后一个节点未必是 $\pi(k)$, $F(\pi_k)$ 为相应的目标值. dynasearch 算法实质上是通过动态规划方法确定 π_k , π_k 可能由以下两种方式产生: 第 1 种方式是将 $\pi(k)$ 直接加到 π_{k-1} 的后边, 即 $\pi_k = (\pi_{k-1}, \pi(k))$, 此种情况不涉及 swap; 第 2 种方式是对 $0 \leq i < k-1$, 先将 $\pi(i+1), \dots, \pi(k)$ 加到 π_i 后边, 即 $(\pi_i, \pi(i+1), \dots, \pi(k))$, 然后交换 $\pi(i+1)$ 和 $\pi(k)$ 的位置, 从而得到 $\pi_k = (\pi_i, \pi(k), \pi(i+2), \dots, \pi(k-1), \pi(i+1))$. 为了方便描述第 2 种方式, 引入下列符号: $\pi_i = (\pi_i(0), \pi_i(1), \dots, \pi_i(i))$, $0 \leq i \leq k-1$, $\pi_i(l)$ 表示对应的部分最优排序中第 l 位置上的子批. 记交换 $\pi(k)$, $\pi(i+1)$ 的位置后得到的新的排序的相关符号为

$$\tilde{\varphi}_{\pi(l)}^{i+1,k} = \begin{cases} \varphi_{\pi(k)}, & l = i+1; \\ \varphi_{\pi(i+1)}, & l = k; \\ \varphi_{\pi(l)}, & l \neq i+1, k. \end{cases}$$

$$\tilde{u}_{l,\pi}^{i+1,k} = \sum_{s=l}^n \tilde{\varphi}_{\pi(s)}^{i+1,k}, \quad l, k = i+1, \dots, n.$$

其中: $\tilde{u}_{l,\pi}^{i+1,k}$ 表示对于给定排序 $\pi = (\pi(0), \pi(1), \dots, \pi(n), \pi(n+1))$, 交换第 $i+1$ 和第 k 个位置上对应的元

素后, 第 l 个顶点所对应的权重. 至此, dynasearch 算法的动态规划算法可按以下步骤进行计算:

Step 1: 初始条件

$$F(\pi_0) = 0, \quad F(\pi_1) = u_{\pi}^0 d_{\pi(0)\pi(1)}. \quad (39)$$

Step 2: 对于 $k = 2, 3, \dots, n$, 递推公式可表示为

$$F(\pi_k) = \min \{F(\pi_{k-1}) + u_{\pi}^k d_{\pi_{k-1}(k-1), \pi(k)}, \min_{0 \leq i < k-1} \{F(\pi_i) + L(i+1, k)\}\}. \quad (40)$$

其中对任意的 $0 \leq i < k-1$, 有

$$L(i+1, k) = \tilde{u}_{i,\pi}^{i+1,k} d_{\pi_i(i), \pi(k)} + \tilde{u}_{i+1,\pi}^{i+1,k} d_{\pi(k), \pi(i+2)} + \sum_{l=i+2}^{k-2} \tilde{u}_{l,\pi}^{i+1,k} d_{\pi(l), \pi(l+1)} + \tilde{u}_{k-1,\pi}^{i+1,k} d_{\pi(k-1), \pi(i+1)}.$$

这里的 $\pi_i(i)$ 表示部分排序 π_i 中排在最后一个位置 i 所对应的子批.

3.3 不可行解修复策略

约束(16), (17)被松弛到目标函数中, 经过拉格朗日松弛后得到的解不一定可行, 因此需要对其进行修复. 将调度子问题 \tilde{P}_{SQ} 的最优排序解 $\pi^* = (\pi^*(0), \pi^*(1), \dots, \pi^*(n), \pi^*(n+1))$ 代入式(18)得到各子批的完工时间 $C_{\pi(k)}$, 然后由式(16), (17)分别计算 $E_{\pi(k)}$ 和 $T_{\pi(k)}$ 的值.

3.4 拉格朗日对偶问题求解策略

由于拉格朗日函数是不可微、分段线性凹函数, 因此常用次梯度法(SM)解拉格朗日对偶问题. 标准次梯度法在拉格朗日乘子更新过程中可能会产生搜索方向发生所谓的“Z字形”现象, 造成收敛不稳定、速度慢的结果^[20]. the bundle method(BM)^[24]及 the volume method(VM)^[25]可以克服这一弱点, 但 BM 和 VM 为了获取较好的搜索方向, 需要付出大量运算负担^[26]. 文献[27]对 SM 进行了改进, 提出均衡方向策略(ADS), ADS 在解拉格朗日对偶问题时具有运算速度快且可以获得较好结果的优点.

标准 SM 与 ADS 的区别在于第 p 次迭代时的搜索方向 d^p 的差异. 在每次迭代时, 标准 SM 的搜索方向只考虑当前的次梯度方向, 即 $d^p = g^p$, 其中 g^p 表示第 p 次迭代时的次梯度. ADS 不仅考虑了当前第 p 次迭代时的次梯度方向, 还结合了前一次迭代时的搜索方向, 即在 ADS 中

$$d^p = g^p + \eta^p d^{p-1}, \quad (41)$$

其中 $\eta^p \geq 0$ 为当前迭代时搜索方向与次梯度偏离系数, 且 $\eta^p = \|g^p\| / \|d^{p-1}\|$ (规定 $d^0 = 0$).

记第 p 次迭代时拉格朗日函数 $L(\lambda, \mu)$ 在 (λ^p, μ^p) 处的次梯度为

$$\mathbf{g}^p = (g_{11}^p, \dots, g_{1n}^p, g_{21}^p, \dots, g_{2n}^p),$$

其中

$$\begin{aligned} g_{1i}^p &= C_i^p - R_i - T_i^p, \\ g_{2i}^p &= R_i - C_i^p - E_i^p, \quad i = 1, 2, \dots, n. \end{aligned} \quad (42)$$

对应的搜索方向为

$$\mathbf{d}^p = (d_{11}^p, \dots, d_{1n}^p, d_{21}^p, \dots, d_{2n}^p).$$

拉格朗日乘子按如下方式更新:

$$\begin{aligned} \lambda_i^{p+1} &= \max \{0, \lambda_i^p + \tau^p d_{1i}^p\}, \\ \mu_i^{p+1} &= \max \{0, \mu_i^p + \tau^p d_{2i}^p\}, \quad i = 1, 2, \dots, n, \end{aligned} \quad (43)$$

τ^p 表示乘子第 p 次更新时的步长. 一般情况下, 步长的计算方法为

$$\tau^p = \zeta^p \frac{Z_{UB} - L(\lambda^p, \mu^p)}{\|\mathbf{g}^p\|^2}. \quad (44)$$

式中: Z_{UB} 是直到第 p 次迭代时所获得的拉格朗日松弛问题解的最好上界, 可以是原问题的一个可行解; ζ^p 为步长调节参数, $0 < \zeta^p \leq 2$, 如果在给定的迭代次数内, 上下界没有改进, 则改变 ζ^p 的值. 若本文连续 50 次迭代都没有使解的质量得到改善, 则 $\zeta^{p+1} = 0.8\zeta^p$. 算法的终止条件为:

- 1) 达到预先给定的最大迭代次数.
- 2) 对偶间隙 GAP 满足

$$\text{GAP} = \frac{Z^* - L(\lambda^*, \mu^*)}{L(\lambda^*, \mu^*)} < \varepsilon. \quad (45)$$

其中: ε 为一充分小的正数, Z^* 和 $L(\lambda^*, \mu^*)$ 分别为迄今为止原问题目标值的最好上界及最好的拉格朗日松弛解.

拉格朗日乘子 (λ_i, μ_i) 的初始值的选取, 对均衡方向策略求解拉格朗日对偶问题的性能无显著影响^[28]. 因此, 一般情况下, 对拉格朗日乘子初始值的选取不加以限制, 但本文中的拉格朗日乘子 (λ_i, μ_i) 的选取有如下性质:

性质 1 在拉格朗日对偶问题 LD 的最优解中, 对任意的 i , 拉格朗日乘子 (λ_i, μ_i) , 满足: $w_i \geq \lambda_i$, $e_i \geq \mu_i$.

证明 若存在 $1 \leq i \leq n$, $w_i < \lambda_i$ 或 $e_i < \mu_i$, 由 3.2 节中的子问题 P_E 及 P_T 的解法, 有 $E_i = \infty$ 或 $T_i = \infty$. 此时有 $L(\tilde{\lambda}, \tilde{\mu}) = -\infty$, 其中 $\tilde{\lambda}, \tilde{\mu}$ 为包含 λ_i, μ_i 的拉格朗日乘子向量. 由于拉格朗日对偶问题是对拉格朗日函数 $L(\lambda, \mu)$ 求极大值问题, 对偶问题的最优解中对应的乘子中不会出现 $w_i < \lambda_i$ 及 $e_i < \mu_i$. \square

因此, 在解拉格朗日对偶问题时, 只需要考虑 $w_i \geq \lambda_i$, $e_i \geq \mu_i$ 的情形.

4 实验结果

为了验证模型及算法的有效性, 将本文开发的算法(记为 LR)与文献[6]的算法进行了比较, 实验数据的生成方法与[6]相同. 具体来说, 本文采用[6]中所提出的思想, 构造了两种启发式算法: 交替 Tabu 搜索算法(ATS)和串行式 Tabu 搜索算法(STS), 来求解模型 MILP. 以上所有 3 种算法均在 Matlab 7.0 环境下进行仿真.

以某汽配车间的一条同步装配线为背景进行仿真实验. 该焊装车间在一个计划周期(4 800 min, 10 个工作日)内, 需要装配 5 种汽车, 其需求量分别为 40, 50, 140, 200, 50 辆. 式(11)~(18)中的其他参数已经给定, 其中装配时间 p_{ij} 是一个服从正态分布的随机变量, 方差是其均值的平方乘以 0.002 5, 即 $p_{ij} \sim N(\bar{p}_{ij}, 0.0025\bar{p}_{ij}^2)$. 装配线有 33 个工位, 随机产生 10 组数值例子进行仿真试验.

表 1 给出了 3 种算法的实验结果, 算法的性能以仿真时间及解的平均改善率来衡量. 其中每组数值实验中解的改善率定义为 $S_i = (Z_i - Z_{LR})/Z_{LR} \times 100\%$, Z_i 代表算法 ATS 或 STS 的目标值, 解的平均改善率取 10 次仿真结果改善率的平均值. 从表 1 的比较结果可以看出, STS 算法所用的仿真时间最少, ATS 算法的平均目标值比 STS 算法低 0.99%, 而 LR 算法的平均目标值最小, 仿真时间比 STS 算法多 5.17 s, 但是在可以接受的范围内.

表 1 LR 法与两种禁忌搜索法比较

算法	LR	ATS	STS
平均仿真时间/s	13.58	304.55	8.41
最大仿真时间/s	16.89	344.61	5.47
平均解质量改进率/%	-	1.38	2.26
最大解质量改进率/%	-	5.66	6.10

表 2 给出的实验结果是不同规模下各种算法的仿真结果. 经过预实验发现, 当汽车的种类多于 10 种时, ATS 算法在可接受的时间内不能给出实验结果, 故以下实验不包括 ATS 算法. 表中的 LR1 算法是指前边提到的用标准次梯度法替换均衡搜索方向策略求解拉格朗日对偶问题. 两种拉格朗日松弛算法的最大迭代次数均为 500. 从表 2 提供的实验结果可以得出以下结论:

1) 3 种算法的仿真时间均随着实验规模的增大而增加, 其中 STS 算法增加最显著. 以 30 种汽车实验为例, STS 算法的仿真时间分别是 LR 算法的 3.97 倍和 LR1 算法的 1.77 倍.

表2 3种算法结果比较

汽车种类数	平均仿真时间/s			平均目标值			$(Z - Z_{\min})/Z_{\min} / \%$		
	LR	LR1	STS	Z_{LR}	Z_{LR1}	Z_{STS}	LR	LR1	STS
5	13.58	35.31	8.41	1.679 9e+005	1.668 2e+005	1.723 1e+005	0.70	0.00	3.29
10	25.77	89.84	38.74	1.784 3e+005	1.788 6e+005	1.932 4e+005	0.00	0.24	8.30
15	58.56	207.23	114.79	2.968 0e+005	2.957 4e+005	3.329 8e+005	0.36	0.00	12.59
20	214.41	420.11	592.38	4.680 0e+005	4.674 7e+005	5.363 8e+005	0.11	0.00	14.74
25	290.16	645.56	814.33	8.212 8e+005	8.215 7e+005	9.410 1e+005	0.00	0.04	14.57
30	481.23	1 083.71	1 914.41	7.586 1e+005	7.705 9e+005	9.148 4e+005	0.00	1.57	20.59

2) 对于同种规模的问题, LR1 算法所消耗的仿真时间是 LR 算法的 2~3 倍。以 15 种及 30 种汽车为例, LR1 算法的仿真时间分别是 LR 算法的 3.53 及 2.25 倍, 其原因是前面提到的标准次梯度法在求解拉格朗日对偶问题时, 容易产生“Z 字形”现象, 从而使算法的收敛速度变慢。

3) 3 种算法中, 两种拉格朗日松弛法具有较好的计算结果。它们的平均目标值改进率的平均值分别比 STS 算法改进 12.1%, 12.0%。采用 ADS 策略的 LR 比 LR1 算法的平均目标值改进率平均改进 0.69%。

图 3 反映了 LR 和 LR1 两种拉格朗日法中, LR 法的平均迭代次数较小, 所用的仿真时间也较少。

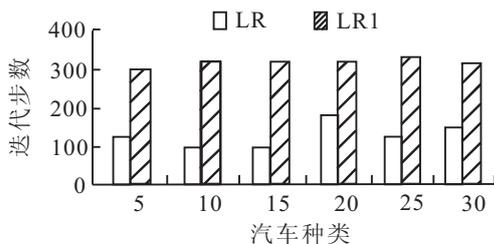


图3 两种拉格朗日法迭代次数比较

图 4 比较了求解两种拉格朗日对偶问题时, 算法停止时的平均对偶间隙 (GAP)。从图 4 可以看出, LR 法的平均 GAP < 3%, 表明 LR 法解的质量比较理想。而 LR1 法的平均 GAP 波动较大, 当问题的规模较大时, LR1 法的平均 GAP 增加显著。

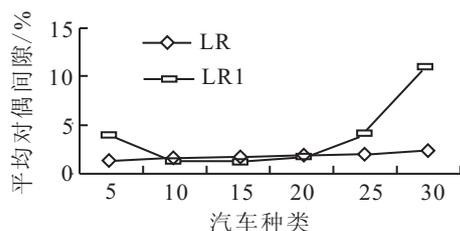


图4 两种拉格朗日法平均GAP比较

5 结 论

本文讨论了汽车同步装配车间的生产计划与调度集成优化问题, 构建了整体考虑调度属性与批量约束的混合整数规划模型, 通过拉格朗日松弛法将调

度与计划的耦合约束解耦, 并将原问题分解成批量计划、调度等子问题。由于调度子问题的特殊性, 将其转化为一类特殊形式的与时间相关的旅行商问题, 并构造了 dynasearch 算法求解。通过与文献 [6] 所提出的方法相比, 验证了本文方法的有效性。

参考文献(References)

- [1] Lasserre J B. On the importance of sequencing decisions in production planning and scheduling[J]. Int Trans on Operational Research, 2003, 9(6): 779-793.
- [2] Lasserre J B. An integrated model for Job-Shop planning and scheduling[J]. Management Science, 1992, 38(9): 1201-1211.
- [3] Shobrys D E, White D C. Planning, scheduling and control systems: Why cannot they work together[J]. Computers and Chemical Engineering, 2002, 26(2): 149-160.
- [4] Maravelia C T, Sung C. Integration of production planning and scheduling: Overview, challenges and opportunities[J]. Computers & Chemical Engineering, 2009, 33(12): 1919-1930.
- [5] 周泓, 谭小卫. 一种两层生产计划问题建模及其遗传算法设计[J]. 系统仿真学报, 2007, 19(16): 3643-3648. (Zhou H, Tan X W. Integrated model for two-stage production planning and its genetic algorithm design[J]. J of System Simulation, 2007, 19(16): 3643-3648.)
- [6] Yan H S, Xia Q F, Zhu M R, et al. Integrated production planning and scheduling on automobile assembly lines[J]. IIE Transactions, 2003, 35(8): 711-725.
- [7] 张晓东, 严洪森. 一类 Job-Shop 车间生产计划与调度的集成优化[J]. 控制与决策, 2003, 18(5): 581-584. (Zhang X D, Yan H S. Integrated optimization of production planning and scheduling for a kind of job-shop[J]. Control and Decision, 2003, 18(5): 581-584.)
- [8] Yan H S, Zhang X D. A case study on integrated production planning and scheduling in a three-stage manufacturing system[J]. IEEE Trans on Automation Science and Engineering, 2007, 4(1): 86-92.

- [9] 周泓, 王建, 谭小卫. 一种求解集成生产计划的混合协同进化算法[J]. 计算机集成制造系统, 2007, 13(7): 1412-1418.
(Zhou H, Wang J, Tan X W. Hybrid collaborative EA for integrated production planning[J]. Computer Integrated Manufacturing Systems, 2007, 13(7): 1412-1418.)
- [10] 尚文利, 范玉顺. 成批生产计划调度的集成建模与优化[J]. 计算机集成制造系统, 2005, 11(12): 1663-1667.
(Shang W L, Fan Y S. Integrated optimization modeling of production planning and scheduling for batch production[J]. Computer Integrated Manufacturing Systems, 2005, 11(12): 1163-1167.)
- [11] 熊锐, 陈浩勋, 胡保生. 一种生产计划与车间调度的集成模型及其拉氏松弛求解法[J]. 西安电子科技大学学报, 1996, 23(4): 509-516.
(Xiong R, Chen H X, Hu B S. An integration model for production planning and job shop scheduling and its Lagrangian relaxation-based solution approach[J]. J of Xidian University, 1996, 23(4): 509-516.)
- [12] Conway R W, Maxwell, McClain W, et al. The role of work-in-process inventory in serial production lines[J]. Operations Research, 1988, 36(2): 229-241.
- [13] Soylu B, Kirca O. Flow shop-sequencing problem with synchronous transfers and makespan minimization[J]. Int J of Production Research, 2007, 45(15): 3311-3331.
- [14] Eliiyi D T, Ozlen M. A Lagrangean relaxation approach for the mixed-model flow line sequencing problem[J]. Computers & Operations Research, 2008, 35(3): 933-943.
- [15] Meyr H. Simultaneous lotsizing and scheduling on parallel machines[J]. European J of Operations Research, 2002, 139(2): 277-92.
- [16] Jozefowska J, Zimniak A. Optimization tool for short-term production planning and scheduling[J]. Int J of Production Economics, 2008, 112(1): 109-120.
- [17] 张根保, 黄强. 对多品种小批量生产方式的反思[J]. 四川兵工学报, 2005(4): 3-6.
(Zhang G B, Huang Q. Thinking over the production mode of many varieties and small batch[J]. Sichuan Ordnance J, 2005(4): 3-6.)
- [18] 潘常春, 杨根科, 孙凯, 等. 带最小批量约束的计划问题及其拉格朗日松弛算法[J]. 控制理论与应用, 2009, 26(2): 133-138.
(Pan C C, Yang G K, Sun K, et al. A Lagrange relaxation algorithm for capacitated lot-size problem (CLSP) with minimum lot-size constraint[J]. Control Theory & Applications, 2009, 26(2): 133-138.)
- [19] Swoveland C. A deterministic multi-period production planning model with piecewise concave production and holding-backorder cost[J]. Management Science, 1975, 21(9): 1007-1013.
- [20] 轩华, 唐立新. 实时无等待HFS调度的一种拉格朗日松弛算法[J]. 控制与决策, 2006, 21(4): 376-389.
(Xuan H, Tang L X. Lagrangian relaxation algorithm for real-time hybrid flowshop scheduling with no-wait in process[J]. Control and Decision, 2006, 21(4): 376-389.)
- [22] Congram R K. Polynomially searchable exponential neighbourhoods for sequencing problems in combinatorial optimization[D]. UK: University of Southampton, 2000.
- [22] Congram R K, Potts C N, Velde S L. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem[J]. INFORMS J on Computing, 2002, 14(1): 52-67.
- [23] 唐立新, 赵任. 强化Dynasearch & TS算法求解酸轧生产调度问题[J]. 自动化学报, 2010, 36(2): 304-313.
(Tang L X, Zhao R. A new enhanced-dynasearch & ts for the pickling-rolling scheduling problem[J]. Acta Automatica Sinica, 2010, 36(2): 304-313.)
- [24] Zhang D, Luh P B, Zhang Y. A bundle method for hydrothermal scheduling[J]. IEEE Trans on Power Systems, 1999, 14(4): 1355-1361.
- [25] Barahona F, Anbil R. The volume algorithm: producing primal solutions with a subgradient method[J]. Mathematical Programming, 2000, 87(3): 385-399.
- [26] Rong A Y, Lahdelma R, Luh, P B. Lagrangian relaxation based algorithm for trigeneration planning with storages[J]. European J of Operational Research, 2008, 188(1): 240-257.
- [27] Sherali H D, Ulula O. Conjugate gradient methods using quasi-Newton updates with inexact line searches[J]. J of Mathematical Analysis and Applications, 1990, 150(2): 359-377.